

# Modeling Equipment Hierarchy and Costs for ICT solutions

Jonathan Spruytte\*, Marlies Van der Wee, Sofie Verbrugge, Didier Colle  
IDLab, Department of Information Technology at Ghent University – imec,  
Technologiepark-Zwijnaarde 15, 9052 Gent, Belgium  
[jonathan.spruytte@ugent.be](mailto:jonathan.spruytte@ugent.be)

**Abstract**—In the early 2000s, a large number of companies thrived mainly thanks to the fast-paced evolution of network and Internet technologies. A similar trend is now emerging with the rise of the Internet of Things (IoT), using which almost every *thing* can be part of the Internet. Both groups of companies have important ICT networks as their core assets. In order to validate the feasibility of the business models of such companies, the relevant costs and revenues should be modeled. This publication focuses on the relevant costs, which can be divided into two categories: process costs and equipment costs, the latter being the focus here.

For equipment costs, no formal standard exists. As a result, most studies make use of use case-specific ad hoc models (typically a combination of visualization and spreadsheet modeling), which tend to be error-prone as well as hard to understand and reuse. To solve these issues, we developed the Equipment Coupling Modeling Notation (ECMN), which allows for both visualization and calculation while focusing on simplicity, flexibility and reusability. ECMN is a flowchart-like notation based on a small number of building blocks, which allows for hierarchical modeling by means of nesting models (using submodels).

In this study, ECMN was applied to an IoT use case to show its strengths, based on which a comparison was made with various ad hoc models using a set of requirements.

**Index Terms**—Techno-economics, equipment costs, cost model, equipment hierarchy, Equipment Coupling Modeling Notation (ECMN).

was not yet launched (2004) and Google started having its first successes. Now, in 2018, an entirely different group of companies is starting to emerge thanks to the popularity of the Internet of Things (IoT), using which almost every *thing* can be part of the Internet. Typical examples are found in connected homes: our fridge may text us when the milk has gone bad, and our heating may start up as soon as it detects we have left the office. IoT does not only simplify our personal life; it allows businesses to transform or enhance their existing business model as well as for new IoT-centric business models to arise. A variety of examples can be found in digital health (e-Health), smart transport (fleet monitoring, smart parking systems), smart buildings (smart control of lightning) and manufacturing (smart factories monitoring every piece of equipment).

In order to evaluate the feasibility of any business model (of either newly formed companies or companies undergoing substantial changes), both the expected revenues and the expected costs should be modeled in detail. Modeling the revenue of a business strongly depends on the type of business and is considered out of scope for this publication. Costs, on the other hand, are closely linked to technology and can be categorized as follows: on the one hand, there are equipment costs typically expressed as a list of required equipment elements represented in a Bill of Materials (BOM), and, on the other hand, there are process-based costs which originate from (non-trivial) internal processes. Note that process cost modeling is not considered in this publication.

As is shown in the next section, there is currently no standard available for equipment cost modeling. This publication proposes a generic notation for modeling and calculating the cost of equipment named Equipment Coupling Modeling Notation (ECMN). ECMN combines equipment properties (unit costs, lifespan, power usage, etc.) with any possible relationships between pieces of equipment (e.g. a server demands a slot in a rack, a corridor requires an access point every 20 meters, etc.) to get a detailed overview of the total cost of the equipment (listed as a BOM) and a reliable estimation of the Total Cost of Ownership (TCO), including the investment and

## 1 MODELING EQUIPMENT COST, AN ESSENTIAL PART OF BUSINESS MODELING

Nowadays, many new companies mainly exist because of the fast-evolving nature of network- and Internet-related technologies. Back in 2002, Netflix was still shipping DVDs, Amazon only sold books, Facebook

operational cost such as energy, maintenance and replacement costs.

The remainder of the paper is structured as follows: a number of possible approaches to equipment cost modeling are discussed in section 2. After introducing ECMN in section 3, we propose, in section 4, an equipment model for a smart cow monitoring system as well as three additional use cases from a more high-level perspective. Section 5 compares ECMN with the ad hoc models discussed in section 2. Finally, in section 6, we summarize and list a number of potential future steps.

## 2 MODELING EQUIPMENT COST

When looking at cost modeling (and modeling in general), a typically main distinction that is made is top-down vs. bottom-up. Using a top-down approach, the problem at hand is being broken down in smaller sections. Top-down models put initial focus on defining the high-level architecture and add detail in additional refine steps. Bottom-up approaches work differently, these start by modeling the smallest levels in detail and build up to higher-level often ending up in more detailed and optimized solutions. In a network context, a top-down model would start from the (existing) network, drilling it down all the way up to the means of how users should get access. In a bottom-up approach, the starting point would be modeling the user and its technical requirements and from there on, the network would be modeled in a way these requirements can be covered.

Besides the choice of modeling approach, the required level of detail should be chosen. For example, in a network setting, will the deployment be modeled using

geographical (GIS) information or will users (and homes) be abstracted?

Furthermore, whether the intended outcome of the study are the estimated costs or the developed cost model itself, makes a great difference as well. If the results of the study are the main goal, very specific models (e.g. technology) and tools (e.g. vendor-specific) can be applied. On the other hand, if the goal is to develop a model which can be applied in various situations (e.g. other use cases or other technologies), it is more important to focus on using a generic approach.

Lastly, various methods for expressing cost are available as well: using fractional models, (small) costs are expressed as a relation to other costs. For example, maintenance cost can be expressed as a percentage of the upfront cost. Using driver-based modeling, a small number of cost drivers are identified which drive the cost of the model at hand. Typical cost drivers are the number of users or homes to be connected.

Practical steps for planning a network deployment as well as more details about network equipment cost modeling are discussed in <sup>1</sup>.

While currently there is no standard available for equipment cost modeling, the literature does contain a variety of cost models. Among the large number of relevant publications, two main types of studies can be discerned: optimization studies, which attempt to optimize a part of the cost of the corresponding hardware; and bottom-up models, which calculate or estimate the cost of a set of equipment or new network roll-out, based on a number of cost drivers.. The objective of ECMN is to improve upon the latter and simplify the notion of hierarchy in a model.

**Table 1: Overview of studies with a clear equipment cost modeling component.**

Reference	Focus O=Optimization C=Cost Analysis	Type of visualization	Level of technical detail	Cost information in representation	Cost approach T=Technical parameters C=Typical cost drivers
Pedrola <sup>2</sup>	Optimization	Technical	High	None	Technical parameters
Rambach <sup>3</sup>	Cost analysis	Conceptual	Low/Medium	None	Technical parameters
Gunkel <sup>4</sup>	Cost analysis	Conceptual	Low	Relative cost units	Technical parameters
Chuan <sup>5</sup>	Optimization	Conceptual	High	None	Technical parameters
Rokkas <sup>6</sup>	Cost analysis	Conceptual	Low	None	Typical cost drivers
Abbas <sup>7</sup>	Optimization	Topology	Low	None	Technical parameters
Schneir <sup>8</sup>	Cost analysis	Conceptual	Medium	None	Typical cost drivers
Tsilipanos <sup>9</sup>	Cost analysis	None	N/A	N/A	Typical cost drivers
Araújo <sup>10</sup>	Optimization	Topology	Low	None	Technical parameters
Mahloo <sup>11</sup>	Cost analysis	Conceptual	Low	None	Technical parameters
Martínez <sup>12</sup>	Cost analysis	Conceptual	Low	None	Typical cost drivers
Skaljo <sup>13</sup>	Optimization	Conceptual	Medium	None	Technical parameters

Troulos <sup>14</sup>	Cost analysis	Conceptual	Low	None	Typical cost drivers
Boone <sup>15</sup>	Cost analysis	None	N/A	N/A	Typical cost drivers
Lang <sup>16</sup>	Optimization	Conceptual	Low	None	Technical parameters
Werner <sup>17</sup>	Optimization	None	N/A	N/A	Technical parameters
Werner <sup>18</sup>	Cost analysis	None	N/A	N/A	Technical parameters
Machuca <sup>19</sup>	Cost analysis	Conceptual	Low	None	Typical cost drivers
Koomey <sup>20</sup>	Cost analysis	None	N/A	N/A	Typical cost drivers
Leiva <sup>21</sup>	Cost analysis	Conceptual	High	None	Technical parameters
Chiha <sup>22</sup>	Cost analysis	Conceptual	Low	None	Typical cost drivers
Schneir <sup>23</sup>	Cost analysis	Conceptual	Low	None	Typical cost drivers

The main disadvantage of the existing models as listed in Table 1 is that the visual representation and the actual mathematical calculations are two separate parts. Having to model the same problems twice obviously increases the total time required to model the problem, but it also risks introducing inconsistencies between both parts. Having two separate models also complicates sharing work with other parties as well as (internal) reuse. For the remainder of this publication, we will refer to this combined approach as ‘ad hoc modeling’.

Table 1 reveals two types of visualizations are mainly used: conceptual and technological. The former are typically made in generic drawing tools (e.g. Visio), while the latter are mostly created in technology/vendor-specific tools (e.g. Cisco Modeling Labs). For the actual cost analysis, one typically falls back to spreadsheet or spreadsheet-like tools.

Spreadsheet modeling is the generic term for using spreadsheet software to model pretty much anything: ranging from modeling linear wear impact on charge motion in tumbling mills<sup>24</sup> to the analysis of the groundwater level rise problem in Jeddah (a Saudi Arabian port city)<sup>25</sup> and financial planning<sup>26</sup>. Spreadsheets offer a generic solution for a large variety of problems, even though the strength and the capabilities of each of the created models strongly depend on the user performing the modeling task. At the same time, it is the users who are the source of most errors or inefficiencies: 37.1% of the users admit to always starting from an empty model instead of re-using an existing design or template; 31.9% indicate that they only sometimes test a model (e.g. testing extreme cases, testing results for plausibility, validating used formulas), while 17.1% even admit to never testing a model at all.<sup>27</sup> Additionally, up to 25% of the respondents are entirely unaware of the risks of errors in spreadsheets, and as little as 11.5% of the created spreadsheets are only used by a single user, confirming the need for clear, easy-to-

understand and easy-to-reuse approaches. However, the problem does not solely lie with the users, as 60% of users reveal that their company has no formal standards when it comes to spreadsheets, while only a lucky 35% have some informal guidelines to follow. As mentioned before, spreadsheets offer a generic solution, but in combination with a lack of a formal approach, there are many things that can go wrong, such as, wrongly used functions, misinterpretation of output, copy/paste errors or wrongly re-using previous spreadsheets.<sup>28</sup> These kinds of errors can have severe consequences because “*errors can lead to poor decisions and cost millions of dollars.*”.<sup>29</sup> In other words, there is an apparent need for a combination of visualization and reliable cost calculation, which will be introduced and argued for in this paper.

In addition to purely academic approaches, there are also various tools (commercial, free or even open source) available which can be linked to equipment modeling of ICT networks. However, the objectives and key parameters of these tools are wide-ranging and diverse, as shown in Table 2. Comparable to the academic literature in the section above, one of the key differences of the listed tools is the modeling level. Some tools offer a generic network modeling solution while others have a more narrow scope (technology or even vendor specific). Additionally, while some tools have as focus the modeling and simulation of existing or new networks, others rather focus upon the validation (e.g. is an area fully covered wirelessly) of networks. Some tools also offer fully automated approaches. These allow the users to provide some input (e.g. geographical input and corresponding configuration parameters) resulting in a fully calculated network. This last group of tools typically results in detailed cost information represented in a BOM.

**Table 2: Overview of existing tools related to equipment modeling with their main objectives**

Tool	Description	Main focus:	Modeling level:	Automated BOM
------	-------------	-------------	-----------------	---------------

		<ul style="list-style-type: none"> <li>• <b>validation</b></li> <li>• <b>simulation</b></li> <li>• <b>automated modeling/calculation</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>Generic equipment modeling</b></li> <li>• <b>Generic network modeling</b></li> <li>• <b>Technology-specific</b></li> <li>• <b>Vendor-specific</b></li> </ul>	as a result of simulation
Riverbed modeler, part of the Riverbed Steelcenter suite <sup>a</sup>	Discrete event simulation engine for analyzing and designing communication works	Validation and simulation	Generic network modeling	No
NS-3 (improved version of NS-2)	Discrete event simulator for the simulation of IP and non-IP based networks. User focus on Wi-Fi, WiMAX, LTE.	Validation and simulation	Generic network modeling	Possible, if implemented manually
FiberPlanIT	Automatic FTTx network design and deployment planning.	Automated modeling/calculation	Technology: FTTx	Yes
Setics Sttar	Network planning and optimization for FTTx networks	Automated modeling/calculation	Technology: FTTx	Yes
QualNet network simulator software	Planning, testing and training tool that mimics the behavior of a real communications network including Wi-Fi and cellular networks	Validation and simulation	Generic network modeling	No
NetSim	Network and protocol simulation software, including wireless (802.11, LTE, ZigBee, Military Radio)	Validation and simulation	Generic network modeling	No
GNS3	Graphical simulation tool with hardware emulation of multiple vendors (e.g. Cisco, Juniper, Dell)	Validation and simulation	Generic network modeling including vendor specific informatoin	No
OMNeT++	Framework to create network simulators	Not applicable	Framework to create generic networking tools	Possible, if implemented
Vendor-specific models (e.g. Cisco Modeling Labs, eNSP(Huawei))	Virtually building, testing and analyzing networks using a vendor-specific network	Validation and simulation	Vendor specific	/
STEM@ network investment model	Calculating the rollout of various telecommunication networks, linked to expected user and demand growths	automated modeling/calculation	Technology: telecommunication networks	Yes

<sup>a</sup> Previously known as OPNET.

Table 2 is meant to show the variety in the tools rather than provide an exhaustive overview of the available tools. Tools which show no active development, are indicated as no longer maintained or are not publically available, such as GloMoSim, VANETsim, Netkit, and NetXT, have not been included in this list.

As can be seen from the table, these tools are generally focused on network dimensioning instead of generic hierarchical equipment modeling. When looking for tools that are really focused on equipment modeling, we only found very low-level equipment modeling, e.g. Printed Circuit Board (PCB) modeling or microprocessor design. According to our knowledge, no real generic equipment modeling tools exists (besides high-level generic drawing tools such as Visio).

### 3 ECMN, A UNIFORM REPRESENTATION FOR EQUIPMENT COST ESTIMATION MODELS

The modeling approaches described above clearly show that equipment cost modeling is in need of a generic modeling technique that manages to combine the strengths of a visual notation with those of a spreadsheet-based methodology, without containing too many technical details or visualize detailed cost information. This paper proposes a newly developed generic modeling notation, ECMN, specifically designed for the generic cost modeling of equipment. ECMN is a conceptual and technology-independent modeling approach that focuses on simplicity, flexibility and reusability, and combines both visualization and calculation of cost (including a detailed BOM) in a single model. As few technical details are included, technical validation of networks is not the scope of ECMN.

This section introduces the necessary terminology and the modeling notation itself. In section 4, the notation is applied to a set of use cases.

#### 3.1 Terminology

**Equipment cost (estimation) model:** model used to calculate the required equipment (represented as a BOM) and accompanying costs (both upfront and recurring) for a specific scenario (use case), consisting of a set of interlinked cost drivers, equipment and equipment hierarchies.

**Cost driver:** an input parameter which *drives* a change of quantities in the BOM and thus of the cost in a cost model.

**Equipment:** the smallest level of detail considered in the cost model to which accompanying costs (both upfront and recurring) are linked; this can be as big

as an entire data center or as small as the screws to fix a hard disk in a storage system, depending on the level of detail at hand.

#### 3.2 ECMN - Equipment Coupling Modeling Notation

ECMN was originally developed to satisfy the need for easy-to-use and easy-to-reuse equipment models when modeling FTTH networks<sup>30</sup>, but has proven to be more widely applicable. It is a graphical notation which consists of five major building blocks: (sub)models, cost drivers, equipment, aggregators and separators, between which connectors (relations) can be made and configured using granularities (see Table 3). The small set of building blocks, each having a single and clear meaning within a model, results in easy-to-understand cost models.

In ECMN models, each link (connection) between two elements directs a flow of demand from one element to another. These demand flows impose a requirement upon the next element. At the very beginning of each flow, at least one cost driver is required to initiate the demand flow (a model without any drivers will have an empty BOM as a result). Cost drivers are thus the root causes of costs in a business. Typical examples of cost drivers in ICT related problems are the number of customers, the bandwidth required and the number of square meters to be wirelessly covered. These drivers should be considered the input for the model; a model can have as many drivers as required.

Furthermore, every element the demand flow passes can also change the demand flow (aggregators and separators) or add equipment to the BOM:

**Aggregators and separators** allow the use of mathematical functions on incoming demand flows. For example, by multiplying (multiplication is one of the aggregators) the number of customers and the bandwidth per user, the total bandwidth can be used in the model. By using a duplicator (one of the separators), a single demand flow can be used multiple times: for example, each company building requires a number of desks as well as a number of storage servers.



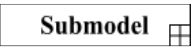



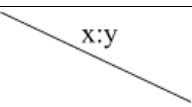
**Equipment** will be added to the BOM based on the incoming demand flow and the applicable granularity. For instance, a connector between the equipment blocks 'server' and 'rack' with a granularity of 21:1 will install 1 rack for each 21 servers.

On top of that, each model can exist on its own or can be linked within another model, meaning that models can

be nested within each other, optimally allowing reuse. Take, for example, a basic cost model of a desk, which requires a table top, four legs and a set of screws. This cost model can exist on its own, or it might be part of the model ‘office’, requiring eight desks and eight desk chairs. In this case, the ‘desk model’ is considered a submodel of the ‘office model’. A submodel can be served by a driver or by an intermediate driver, linked to a parent model (or vice versa).

Finally, all values within the notation have a time component (mathematically speaking  $f(t)$ ), meaning that the values can vary through time. For example, the upfront cost of a piece of equipment can differ year by year. The time component can represent any unit (e.g. minutes, days, years); however, the same unit should to be used for the entire model or set of joined models.

**Table 3: Main building blocks of ECMN**

Icon	Info
	<b>Driver:</b> initiates a single demand flow to the model.
	<b>Equipment:</b> defines a piece of equipment with a set of relevant cost parameters which will be added to the BOM based on the incoming demand flows and corresponding granularities.
	<b>Submodel:</b> is an ECMN model that is linked into another model.
	<b>Intermediate Driver:</b> links a demand flow from a parent model to a submodel or the other way around.
	<b>Aggregator:</b> allows the execution of mathematical functions on one or multiple demand flows (e.g. summing or multiplying demand flows).
	<b>Separator:</b> can split the demand flow into two or multiple flows (based on a mathematical function) or simply duplicate the incoming flow to multiple outgoing demand flows.
	<b>Connector:</b> connects two elements in an ECMN model; a connector can also define granularities (x:y).

As output, two main types are to be considered in an ECMN model:

- The total required amount of each type of equipment (resulting in the BOM), as well as the related total cost of ownership (TCO).
- Any intermediate value within the model contains useful information, e.g. in the second example (Figure 2), the outgoing flow from the SUM-

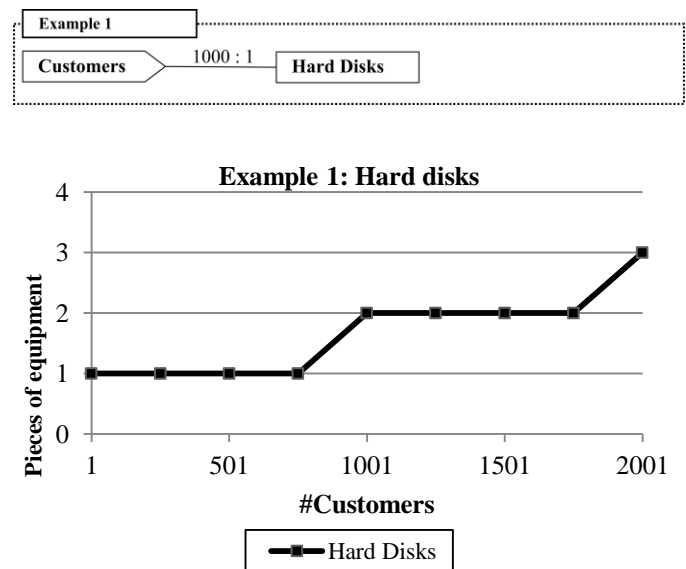
aggregator contains the total number of rack spaces required (per year).

At the time of writing, ECMN has already been published online as a FI-WARE open specification, and we are currently in touch with standardization bodies to translate ECMN into a formal standard. A full definition of the current version of the entire notation, including any updates on the standardization process, is available online at <http://www.technoeconomics.ugent.be/ecmn>.

### 3.3 Modeling using ECMN

In section 4, a use case will be modeled in detail using ECMN. First, we briefly present three small examples to illustrate the five building blocks of ECMN. For each of the examples, the resulting output (in the form of charts) is also included, showing the single cost driver on the x-axis and the corresponding amount of equipment on the y-axis. From these charts, the BOM can easily be extracted.

The first example (Figure 1) might be the most basic equipment model for a cloud storage company, and consists of two interlinked elements: a cost driver and a piece of equipment. In this case, the entire cost consists of a single piece of equipment (Hard Disk), which is driven by the cost driver ‘Customers’: per 1000 customers, 1 hard disk will be installed.



**Figure 1:** The most basic ECMN model consists of a single driver (Customers), connected to a piece of equipment (Hard Disk) using a connector with a 1000-to-1 granularity.

The second example (Figure 2) models the required rack space for a development company. We consider a number of developers (the cost driver); each developer gets a 25% share of a test server for ongoing development (each taking up a single slot in a server

rack). In addition, a storage unit is shared by 1000 developers, which provide daily backups (taking up 4 slots). This example introduces the SUM-aggregator, which adds up both incoming demand flows (representing the required rack space from both the servers and the storage system) and puts the sum on its outgoing connection to the equipment (Racks).

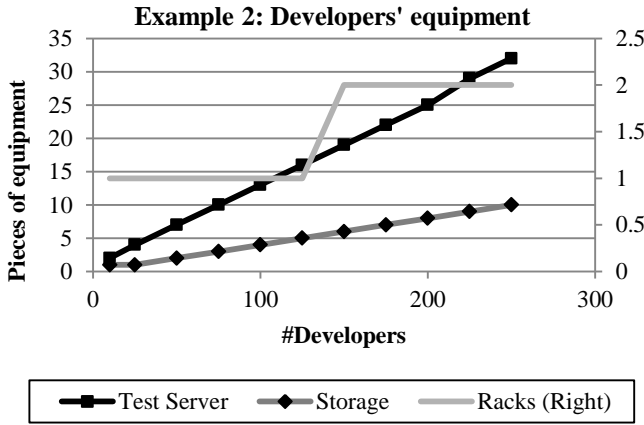
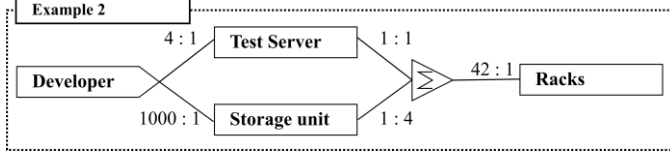


Figure 2: The second exemplary ECMN model consists of a single cost driver, interlinking three pieces of equipment, and demonstrates the use of the SUM-aggregator.

The final example (Figure 3) models a basic IoT network to be installed in the corridor of a large building to monitor a set of parameters (presence of people, temperature differences per floor, etc.). In this example, we introduce submodels and show how these can keep models simple and reusable. The parent model again consists of a single cost driver (Length of Corridor), which is linked to the submodel (with a ratio 10:1) and the equipment (Electricity Cable). This model should be read as “every 10 meters of a corridor, a sensory board is required/installed, and, for each meter of corridor, a meter of cable is required”. The submodel ‘Sensory Board’ then consists of more subcomponents (a Presence Sensor, a Temperature Sensor, an LTE module and a Circuitry Board which groups everything together), and is linked using the intermediate driver ‘#Sensors’. Including submodels is a way to introduce more modeling detail, and to easily replace parts of a model (in this case with another type of sensor node, for example). Replacing a submodel only requires recreating a single link, instead of removing/adding all the required equipment; this leads to much faster results with a reduced chance of errors. Furthermore, when changing the components within the submodel, the new cost elements are automatically included in the parent model.

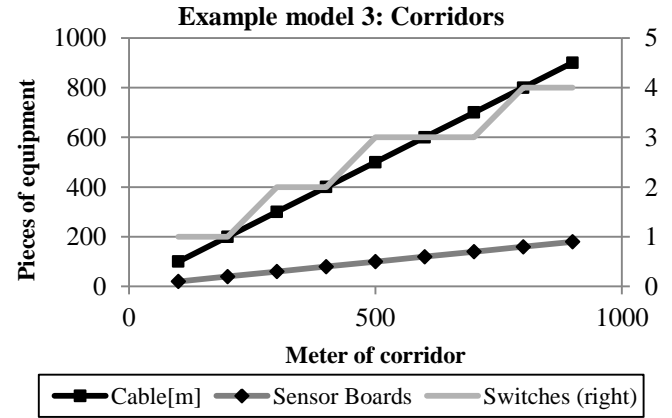
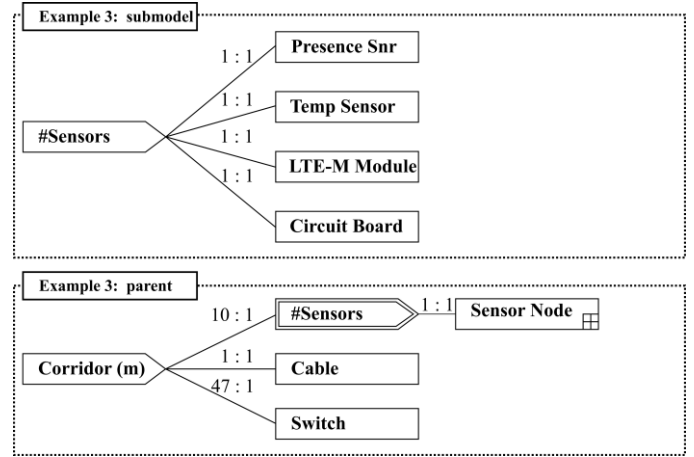


Figure 3: The final example introduces the submodel, interlinked using intermediate drivers, which simplifies the overall model by hiding the most detailed level.

### 3.4 ECMN implemented in the BEMES platform

ECMN represents the modeling notation, that is, the format or the syntax of how an equipment model is built. In order to create actual models, we built an online web interface which provides the functionality for drawing and automatically calculating the BOM and the accompanying costs of a model. This platform is still under construction (the calculation features have not yet been made public at the time of writing), but an initial version is already available online at <http://www.technoeconomics.ugent.be/bemes>. Additionally, the exemplary models which were introduced in the previous section are available at [http://www.technoeconomics.ugent.be/research/papers/2018/ETT\\_spruytte/](http://www.technoeconomics.ugent.be/research/papers/2018/ETT_spruytte/).

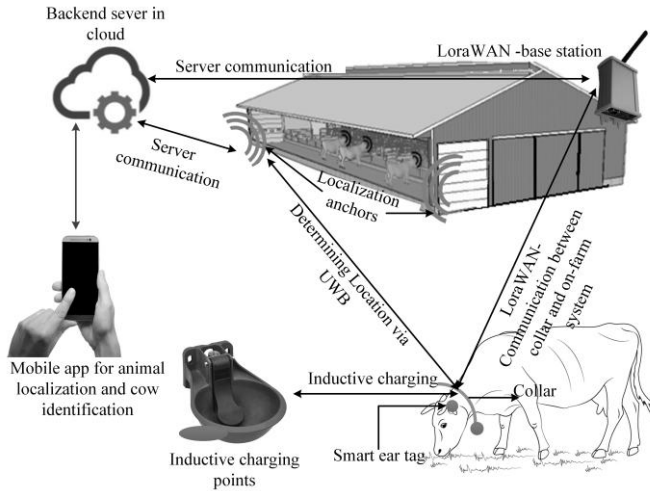
## 4 APPLYING ECMN TO SEVERAL USE CASES

This section applies ECMN to a set of use cases, thus revealing the range of its capabilities. First, a detailed application to an IoT cow monitoring system will show

ECMN’s functionalities and the incremental levels of detail. Subsequently, a couple of other applications are briefly described to demonstrate the flexibility of the modeling notation.

#### 4.1 Modeling a smart cow monitoring system

Closely monitoring livestock is important for various reasons, such as early detection of illness and accurate prediction of fertility. With a growing livestock population per farm, it gets increasingly difficult to keep track of each animal individually. IoT can offer a solution: by providing each animal with a smart ear tag (which contains temperature sensors) and a smart collar (with additional sensors, a GPS module and a communication module), it is possible to collect a considerable amount of data and transmit it to a central monitoring system. This system then aggregates and analyzes the data, and sends out an alert when it detects specific behavioral patterns.



**Figure 4: High-level structural overview of the cow monitoring system, which can roughly be divided into two categories: equipment per cow and equipment per farm.**<sup>31</sup>

The wireless data transfer between the collar and the central monitoring system can be implemented using different technological solutions (varying from low-power Wi-Fi, over private mobile networks (3G, 4G) to specific IoT technologies such as LoRaWAN), thus ensuring a constant wireless connection between the cow and the central system. As each animal produces a steady amount of data, storing the data in the collar and offloading it at fixed intervals might not be the best approach. Therefore, each collar requires a constant wireless connection with the central system, preferably both when the animal is inside and when it is outside. The high-level structure of the cow monitoring system is reflected in Figure 4.

The aim of the next few paragraphs is to describe how the modeling of this kind of use case might work, focusing on the different equipment hierarchies, without going into too much detail on the actual costs, the used technologies and the corresponding implementation constraints. We introduce three levels of detail (see Table 4), starting off with just the major building blocks and adding additional detail as we go. This reflects reality, as, when modeling a new business model, not all relevant information is readily available, although some kind of cost estimation is required.<sup>32</sup> All three levels of detail are modeled using ECMN, which allows us to point out the strengths and weaknesses of the developed notation.

**Table 4: The different modeling levels of the Cow Management System, progressively more detailed. For each cost component, it is indicated what kind of cost is expected (U=Upfront, R=Recurring).**

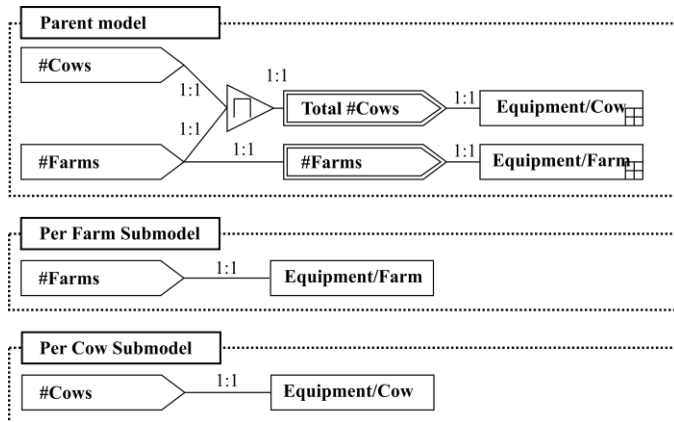
Parameter	Level 1	Level 2	Level 3
Number of cows per farm	Input		
Number of farms	Input		
Total square meter to cover	Input		
Equipment per cow	Undetailed cost per cow (U/R)		
		→	Localization module (U)
		→	Wireless module (U)
		→	Collar (U)
		→	Ear tag (U)



Cow management system	Undetailed cost (U/R)	Charging points 20 per farm (U/R)	
		→	Charging circuitry (U/R)
		→	Communication circuitry (U/R)
		→	Electrical protective circuitry (U/R)
		Cow manager software suite (U/R)	
		Connectivity system (U/R)	
		→	Base stations (U/R)
		→	Cabling (Power)
		→	Cabling (Communication)
		Localization anchors (U/R)	

As the objective was to compare different methods of modeling, not every single detail was modeled for this use case (e.g., ear tag modules and wireless coverage for indoor versus outdoor areas were not included in the model). For the same reason, the cost values were omitted in the different modeling steps (the initial results including the cost values can be found in <sup>31</sup>).

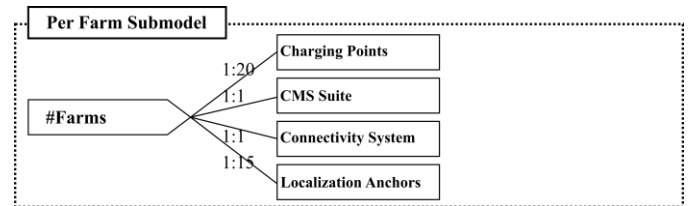
For this specific use case, three levels of detail are introduced, as shown in Table 4. The first level has two inputs that translate into two cost drivers (#Cows Per Farm, #Farms) and two equipment hierarchies (Equipment per Cow and the Cow Management System, CMS). Since the assumption is that more details will be added later on, both the Equipment per Cow and the CMS are modeled in submodels so as not to overcomplicate the main model and for ease of reuse later. For now, these two submodels consist of only a single piece of equipment (representing the undetailed upfront and recurring cost), which is linked into the parent model. The model clearly visualizes that the required equipment per cow depends on both the number of cows and the number of farms (see Figure 5).



**Figure 5: The first modeling step using ECMN consists of two submodels which are linked into a parent model.**

In the second modeling step, more detail is added to the CMS. In order to incorporate this additional information,

the parent model does not have to be altered as the high-level architecture of the cost model remains unchanged. In the CMS submodel (Figure 6), the piece of equipment representing the undetailed cost is removed and four pieces of newly defined equipment are introduced (Charging Points, Cow Manager Software Suite, Connectivity System, and Localization Anchors) and the granularities are updated (e.g., a farm requires 20 charging points). On the off-chance that an error is made in this kind of structure, the error will indubitably be in the submodel (as no changes were made to the other (sub)models), which allows for faster debugging. For the sake of example, we assumed here that no more detail will be added to the CMS. We did this to show the impact of a wrong assumption during modeling.



**Figure 6: The second modeling step using ECMN introduces new pieces of equipment in the CMS submodel, but leaves the parent and other submodel unchanged.**

In the final modeling step, additional information is provided on the equipment per cow, by adding four new pieces of equipment (Localization Module, Wireless Module, Collar and Ear Tag) to replace the undetailed cost per cow. For the CMS, it now becomes obvious that we wrongly assumed that no more detail was going to be added, which can be solved in two ways: a) by introducing four submodels to reflect the four different equipment hierarchies, so that the model can be reused later or in order to keep the hierarchy fairly simple, or b) by adding all the equipment in the submodel CMS, which would only result in a slightly bigger model. The latter is the preferred option when not expecting to ever reuse these parts of the model, which is why it was chosen for this use case. The final resulting model is shown in Figure 7, and can also be consulted online:

[http://www.technoeconomics.ugent.be/research/papers/2018/ETT\\_spruytte/](http://www.technoeconomics.ugent.be/research/papers/2018/ETT_spruytte/).

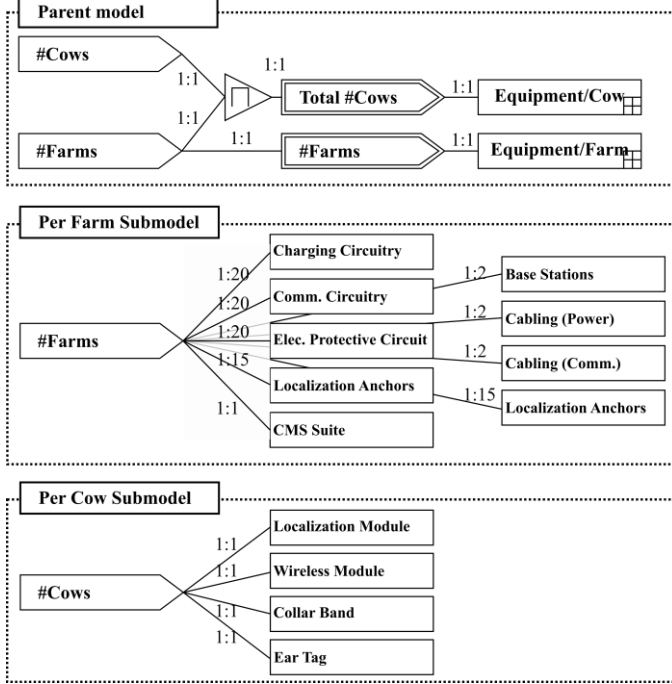


Figure 7: The final modeling step using ECMN adds additional detail to both submodels. The overall structure has remained unchanged through all three modeling steps.

#### 4.2 Modeling additional ICT network-related use cases

This section aims to further establish that ECMN can be used to model equipment in various use cases by providing some additional examples. For these examples, the modeling process is omitted, and only the resulting model is shown. More context regarding these models can be found in the referred paper in each subtitle.

##### 1) Modeling a Cisco ASR 9010 Router<sup>32</sup>

The Cisco ASR 9010 is a modular router in which up to eight line cards can be installed. A line card can hold multiple transceivers to which a single optical feeder is connected. This example (Figure 8) determines how many Cisco ASR 9010 routers are required, based on the incoming number of optical 1, 10, 40 and 100Gbps links.

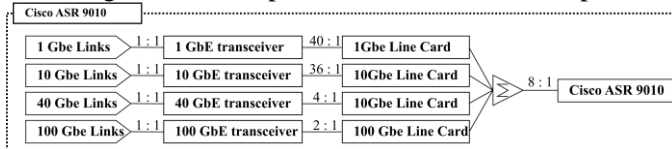


Figure 8: ECMN model of a modular Cisco ASR 9010 Router<sup>32</sup>

##### 2) Modeling a central office for a telecom operator<sup>32</sup>

The second example (Figure 9) models the required number of central offices for a telecom operator based

on the total number of customers. A central office basically creates the connection from the customers' homes (possibly via intermediate street cabinets) to the operator's network.

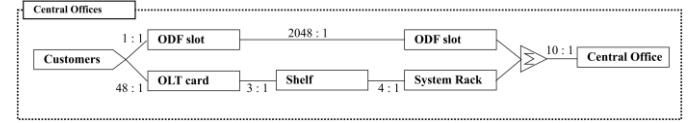


Figure 9: ECMN model of a central office with number of customers as its sole cost driver<sup>32</sup>.

In order to connect the incoming fibers from the end users, Optical Distribution Frame (ODF) racks are installed, which are basically large patch panels with an ODF slot per incoming fiber (customer). In addition, Optical Line Termination (OLT) cards are required, which handle up to 48 incoming fibers (coming from the ODF rack). These OLT cards are installed in shelves, which go into racks. A central office can maximally contain 10 racks (either ODF or system) in total.

##### 3) Modeling the required access points for a Wi-Fi network

The final example (Figure 10) calculates the required number of Access Points (AP) for a Wi-Fi network. This model takes into account two design rules:

- the total area to be covered and the maximal area a single AP can cover as well as
- the maximal number of concurrent users and the maximal number of users a single AP can handle.

The total number of Aps required is the maximum of both design rules.

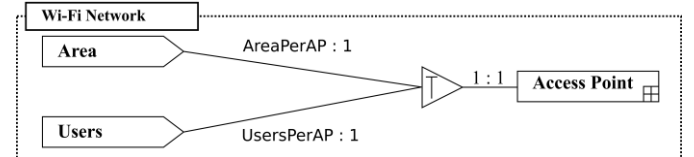


Figure 10: The ECMN model for a Wi-Fi network depends on the area to be covered and the number of concurrent users.

## 5 COMPARISON OF MODELING APPROACHES

In order to compare ECMN with existing ad hoc models, a set of requirements was defined validating different properties. These requirements are based partly on the literature (see literature review in section 2) and partly on our own experience with cost modeling. They are summarized in Table 5 at the end of this section. Where relevant, the visualization and calculation parts of ad hoc models are discussed individually.

### **R1. Level of detail that can be included in the model**

*Which level of detail can be included in the model? Is the level of detail high enough to sufficiently abstract a typical use case?*

ECMN only has a fixed set of cost-related parameters (e.g., a piece of equipment has a price, a lifetime period, a maintenance cost and a size granularity). Other parameters cannot be included. The reason for this is twofold:

- 1) If the parameter is not cost-related, it will unnecessarily increase the size and complexity of the model.
- 2) If the parameter is cost-related, it can usually be modeled as an additional piece of equipment. For example, a piece of equipment (e.g., an Uninterruptible Power Supply, UPS) has a battery which has a specific capacity (and thus a specific price). Although the battery size cannot be included in the equipment in an ECMN model, we can easily incorporate an additional piece of equipment (the battery) with its respective cost parameters and interlink both elements.

While ECMN models use only a small set of predefined elements and parameters (see 3.2 for an overview of the main building blocks), ad hoc models are more flexible (e.g., compare the work of Chuan<sup>5</sup> and Rokkas<sup>6</sup>), as the end user can choose which information to include. As a result, every little detail can be modeled, which has both benefits and drawbacks. Being able to model even the smallest detail can lead to a very accurate model; however, including every piece of information may also result in an unnecessarily complicated model which is more difficult to understand (as discussed in R2). Additionally, unless two models use the exact same structure and building blocks, comparing two models is typically quite a hassle.

### **R2. Level of comprehensibility without (much) additional information**

*Is the model comprehensible without requiring much further information; will an outsider be able to understand the model? Is the representation intuitive? Can information easily be extracted from the model?*

ECMN uses a flow chart-like notation which clearly indicates the relations between elements. Its goal is to be easily understandable by only showing the relevant information, while keeping detailed parameters such as equipment lifetime period hidden from the global view. Because of this graphical approach, ECMN models can easily be used in publications and presentations even if the audience has little to no knowledge of the topic.

The comprehensibility of ad hoc models strongly depends on the type of model. Models created using a typical spreadsheet application *can* be easily understandable and logically (but not visually) structured; however, this solely depends on the technique used and the effort made by the person creating the model. Typical spreadsheet models tend to increase in size and complexity very quickly, resulting in large bulks of data in which a non-informed reader quickly loses overview (e.g., the final tables of the study of Araújo<sup>10</sup>). Furthermore, the visualizations available (large tables of data and complicated charts) are ill-suited to represent the relations between elements. This means that another type of model must be used to visualize the results (doubling the modeling effort). Additionally, making a change in either of the two models means having to carry the change to the other model, thus risking inconsistency errors.

### **R3. Modeling equipment with hierarchical levels**

*Can models easily be built upon each other? Can models be linked into each other or structured in a hierarchical manner?*

As ECMN supports the nesting of (sub)models, it is inherently hierarchical. By means of these submodels, a large cost model can be split into smaller reusable pieces, allowing each model to be calculated either independently or as part of a larger model. This also has a considerable impact on the reusability of ECMN models (see R4). Imagine an IoT device having a sensory board with different types of sensors and an interface board with an LTE module. Using ECMN, both the sensory board and the interface board can be modeled with as many details as needed and afterwards linked into the IoT model. This way, the detailed cost information of each component is present in the submodel and will automatically be included in the total cost calculations, although it is by default hidden from the end user. The IoT model itself can then easily be linked into, for instance, the cost model of an office or a warehouse.

As mentioned in R1, ad hoc models can model any kind of detail, but the level of detail strongly depends on the skills of the person making the model. While creating a visualization which represents multiple, hierarchical levels is easy enough (as shown in Figure 2), calculating these levels using spreadsheets is much more difficult. One possibility is creating a separate model per hierarchical level and linking everything together in an overview sheet. However, linking sheets together to allow for the calculation of multiple values or scenarios requires utmost caution, since a single, incorrectly linked cell can promptly result in inaccurate results.

#### **R4. Ease of reuse of existing models and data**

*Can an existing model easily be reused or recalculated with new values? Can (parts of) the model be copied or linked into another model with little to no overhead?*

ECMN models have a very strict structure, clearly defining the input and output. As a result, it allows external people to rerun a model with new values and little to no any additional information. Reusing (part of) a model is as straightforward as can be. A (part of a) model can easily be incorporated into a larger model by linking it in as a submodel (as mentioned in R3), and output values can be exported back to the parent model for further calculations. Additionally, by linking to an existing model (instead of making a copy), a set of models can depend on the same underlying model. Imagine modeling an LTE receiver for IoT purposes and using it in a number of different models for IoT devices (e.g. a car or a sensory node). When a change is made to the LTE receiver, impacting its cost, the individual costs of the different IoT devices will be automatically adjusted accordingly.

Reusing ad hoc models is typically not as straightforward. Visualization of the model in particular is often use case- or technology-specific (e.g., the work of Leiva<sup>21</sup>) and created in a generic tool (e.g., Microsoft Visio), not focused on a fast reuse of the existing images. Reusing the calculations is in theory simple enough, but can in reality be quite complex. The structures and formats used tend to differ from person to person, which makes interpreting, reusing and merging these models much more difficult (see R2). Moreover, merging changes between different versions of a model may consist of much copy-pasting or may lead to inconsistency issues. Nonetheless, linking data cells from one workbook to another is possible, which allows a user to separate data and functionality and share input values among spreadsheet models. However, sharing formulas is not possible (except for copy-pasting the formula and afterwards editing all the corresponding values), meaning that, typically, the most essential part, the logic, cannot easily be reused.

#### **R5. Calculating the model in a time-oriented fashion**

*Can the model be calculated for multiple periods of time at once, in other words, not changing a time parameter iteratively in order to get new output? Can parameters varying over time easily be defined (e.g., number of customers and energy prices)?*

These questions are irrelevant for the visualization part, so the comparison focuses on the calculation step of equipment cost modeling. Almost every parameter

(except for textual values and values denoting the relations between equipment) within ECMN has a time component (see 3.2 for more details). In other words, every model is by default a time-dependent model. The parameter *t* can represent any kind of time unit (minutes, days, years, etc.), but the same unit must to be used throughout the entire model or set of joined models. Because of this, every ECMN model is inherently time-dependent, meaning that it can easily be used to calculate costs linked to variable inputs such as user adoption, changing prices (e.g. energy prices) and required bandwidth per user (which translates in a higher connection cost in regional, aggregation and core networks). As a direct result, changing the time window of a cost model is only a matter of changing the number of time units (e.g. years) the model should be calculated for.

In order to create time-oriented spreadsheets, there are two common approaches to choose from. The first, and simplest, approach provides a cell ‘time’ which can be adapted by the user and affects all of the relevant functions. However, most analysis will require the user to manually adjust the cell ‘time’ for all relevant values. The second approach uses a column ‘time’, which is then incorporated into the formulas (using the automatic fill functionality). With this approach, users must be vigilant to correctly *anchor* the formulas (using the dollar sign), or risk ending up with incorrect data and hard-to-spot errors to correct. Extending the time-range of a model means having to create or calculate the values of all relevant parameters, which can be time-consuming for a complex model. In addition, if changing the time range of the model was not anticipated and the formulas have not correctly been prepared, the risks discussed above are applicable once again.

#### **R6. Possibility to perform sensitivity analysis (on both the cost drivers and the equipment parameters)**

*Can the sensitivity of a model easily be tested<sup>2</sup>? Can the ranges of the input values easily be defined?*

As in R5, these questions are irrelevant for the visualization part; therefore, the comparison focuses on the calculation part. ECMN itself has no sensitivity capabilities; the BEMES tool (see section 3.4) offers these capabilities. In the BEMES tool, every parameter can be given a set of values, and the model can automatically be calculated for each set of inputs. Afterwards, the tool provides the outputs of every single

<sup>2</sup> Through sensitivity analysis, it is possible to determine how sensitive the output is to changes in the input. As a result, which input has the most impact on the output can easily be detected. This kind of knowledge can afterwards be used in the risk analysis for a business model.

calculation as well as automated statistics. The range of the values can be defined (e.g. a range of linear or exponential steps between two values), can be a predefined list of values or can be calculated automatically (e.g. a 30% range (higher and lower) around the default values with steps of 5%). This way, any type of model can easily be calculated for a wide variety of values, thus greatly simplifying the sensitivity analysis.

The most popular spreadsheet packages usually have some limited capability to perform automated calculations; however, this is typically limited to two parameters because visualizing tables with more than two dimensions is rather difficult. While this approach (measuring sensitivity based on two values) may yield some insights, it cannot be considered sufficient for an extensive model. Alternatively, there are various plug-ins which offer sensitivity analysis functionality such as Oracle Crystal Ball<sup>33</sup> (licensed use) and Life Cycle Costing (LCC)<sup>34</sup> (free to use). These plug-ins may require a certain format, which means that a user has to either consider the right format from the start or spend some time reformatting or even rebuilding the existing model, which may introduce errors.

## R7. Extracting results to include in reports or to serve as input for further calculations

*Can the results of the model easily be exported to be included in further calculations, analysis and reporting? Can the results easily be visualized (e.g. in charts) or shared with other people?*

As mentioned in R6, ECMN itself has no calculation capabilities; these are included in the BEMES tool. After calculation of an ECMN model, BEMES allows the data (all of the data required to create the BOM, as well as the intermediate values of the separators and aggregators) to be presented in dynamically created charts and to be

exported to spreadsheets or comma separated files (csv) using a predetermined fixed format for further analysis. Having a fixed format simplifies this further analysis. Additionally, the BEMES editor also allows for programmatic access (using a REST-interface); this way, the logic and results from ECMN cost models can easily be included in a wide range of simulations (e.g., including the cost of a network node in a network dimensioning algorithm) and analysis (e.g., calculating the impact on the cost of equipment in game-theoretical approaches).

Ad hoc models offer some value when writing reports and publications: using a technology-specific model, as discussed in section 2, allows for a clear interpretation of the relations within an equipment model (much like ECMN does). As these models are typically basic images created in generic tools (e.g. Visio), exporting them is fairly straightforward. When it comes to the calculation of the models using spreadsheets, the results of a model are generally presented alongside the logic or on a separate sheet. These sheets can easily be shared or copied to other locations. However, using them with any programming language may require additional steps (reformatting, exporting to a simple-to-use format (e.g. text or csv)) as well as insider knowledge to successfully interpret the generated file. Converting the results into graphs is typically simple enough, providing that model and results are well structured, as argued in R2.

## Summary requirements

As can be seen from Table 5, ad hoc models definitely have their benefits, even though they typically get their strengths by combining two types of models (visualization and calculation). Through ECMN, we have managed to combine these two functionalities, effectively reaping the benefits of both.

**Table 5: Summary of how well spreadsheet approaches and ECMN match the requirements of cost equipment modeling.**

Requirement	ECMN + BEMES	Ad hoc models
<i>R1: Level of detail</i>	Includes all typical cost parameters	Any level of detail possible, but more detail typically results in a higher complexity
<i>R2: Level of comprehensibility</i>	High	Highly dependent on the structure used by the creator; risk of errors when using separate models for calculation and visualization
<i>R3: Ease of creating hierarchical models</i>	Inherently present by using submodels	For calculation: highly dependent on the structure used by the creator; for visualization: high

		level of ease.
<i>R4: Possibility and ease of reusing models</i>	Inherently present by using submodels	For calculation: highly dependent on the structure used by the creator; for visualization: rarely possible.
<i>R5: Possibility to model in a time-oriented manner</i>	Inherently present	Possible, but error-prone or requiring external plug-ins
<i>R6: Possibility to perform sensitivity analysis</i>	Fully automated using the BEMES editor	Basic built-in capabilities; more functionality only possible by means of external plug-ins
<i>R7: Extraction and visualization of the results</i>	Dynamic charts internally available; possibility to export results in a fixed format to csv for external usage. Has built-in programmatic access to include results in more complex simulations/analysis.	Visualization of results is inherently present in spreadsheets; programmatic use of results requires additional steps such as formatting and writing code to import the results. Visual models can easily be exported as is.

## 6 SUMMARY & FUTURE WORK

Considering the feasibility of a business model requires modeling the (estimated) revenues as well as the (estimated) costs. On the cost side, a distinction is generally made between investment costs, typically expressed as a list of required equipment elements represented in a Bill of Materials, and operational costs linked to (non-trivial) internal processes.

As shown in the literature review (see section 2), no standard exists when it comes to equipment modeling. As a direct result, people tend to fall back on ad hoc modeling, combining two types of models: one for visualization and one for calculation. These models have a large number of drawbacks, such as being error-prone, hard to reuse and often difficult to understand without prior knowledge. For this exact reason, ECMN was developed. ECMN is a conceptual and technology-independent modeling approach. It is a visual, flow chart-like notation, which allows users to visually construct a cost model by interlinking pieces of equipment (including both an upfront cost and a recurring cost) and allowing for additional parameters to define the relations between the equipment. The very core of ECMN consists of five major building blocks, each with a clearly defined goal, thus reducing the overall complexity of the models, resulting in easy-to-understand and reusable models. As a result, ECMN models can easily be shared within teams and externally (e.g., in presentations and publications).

By way of illustration, this paper modeled an IoT use case as well as some introductory example cases using

ECMN. Afterwards, a comparison was made between ECMN and ad hoc modeling approaches, which revealed that ECMN, despite having a limited level of detail, offers a more generic solution to equipment cost modeling. EMCN ensures that models can easily be communicated, shared and reused, which is a strong advantage when compared to the use of ad hoc models and spreadsheet calculations.

At the time of writing, ECMN has already been published online as a FI-WARE open specification, and we are currently in touch with standardization bodies to translate ECMN into a formal standard. The current version of ECMN is available at <http://www.technoeconomics.ugent.be/ecmn>.

In the meantime, we are developing the BEMES web interface, which will allow all interested researchers to create ECMN models and link these cost models into publications, thus simplifying sharing and validating cost models in academic literature and research projects.

## 7 REFERENCES

- 1 Sofie Verbrugge, Koen Casier, Jan Van Ooteghem, Bart Lannoo, White paper: Practical steps in techno-economic evaluation of network deployment planning, published April 14th, 2009, [http://www.technoeconomics.ugent.be/white\\_paper.pdf](http://www.technoeconomics.ugent.be/white_paper.pdf)
- 2 Pedrola, O., Careglio, D., Klinkowski, M., Solé-Pareta, J., & Bergman, K. (2013). Cost feasibility analysis of translucent optical networks with shared wavelength converters. *Journal of Optical Communications and Networking*, 5(2), 104-115.
- 3 Rambach, F., Konrad, B., Dembeck, L., Gebhard, U., Gunkel, M., Quagliotti, M., ... & López, V. (2013). A

- multilayer cost model for metro/core networks. *IEEE/OSA journal of optical communications and networking*, 5(3), 210-225.
- 4 Gunkel, M., Leppla, R., Wade, M., Lord, A., Schupke, D., Lehmann, G., ... & Nakajima, H. (2006, October). A cost model for the WDM layer. In *Photonics in Switching, 2006. PS'06. International Conference on* (pp. 1-6). IEEE.
- 5 Chuan, N. B., Premadi, A., Ab-Rahman, M. S., & Jumari, K. (2010, July). Optical power budget and cost estimation for Intelligent Fiber-To-the-Home (i-FTTH). In *Photonics (ICP), 2010 International Conference on* (pp. 1-5). IEEE.
- 6 Rokkas, T., Neokosmidis, I., Katsianis, D., & Varoutas, D. (2012). Cost analysis of WDM and TDM fiber-to-the-home (FTTH) networks: A system-of-systems approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 1842-1853.
- 7 Abbas, H. S., & Gregory, M. A. (2013, November). Feeder fiber and OLT protection for ring-and-spur long-reach passive optical network. In *Telecommunication Networks and Applications Conference (ATNAC), 2013 Australasian* (pp. 63-68). IEEE.
- 8 Schneir, J. R., & Xiong, Y. (2013). Economic implications of a co-investment scheme for FTTH/PON architectures. *Telecommunications Policy*, 37(10), 849-860.
- 9 Tsilipanos, K., Neokosmidis, I., & Varoutas, D. (2015). Modeling complex telecom investments: A system of systems approach. *IEEE Transactions on Engineering Management*, 62(4), 631-642.
- 10 Araújo, M., & de Oliveira Duarte, A. M. (2011, February). A comparative study on cost-benefit analysis of Fiber-to-the-Home telecommunications systems in Europe. In *Internet Communications (BCFIC Riga), 2011 Baltic Congress on Future* (pp. 65-69). IEEE.
- 11 Mahloo, M., Machuca, C. M., Chen, J., & Wosinska, L. (2013). Protection cost evaluation of WDM-based next generation optical access networks. *Optical Switching and Networking*, 10(1), 89-99.
- 12 Martínez, R. I., Prat, J., Lázaro, J. A., & Polo, V. (2007). A low cost migration path towards next generation fiber-to-the-home networks. In *Optical Network Design and Modeling* (pp. 86-95). Springer, Berlin, Heidelberg.
- 13 Skaljo, E., Hodzic, M., & Mujcic, A. (2015, October). A cost effective topology in fiber to the home point to point networks based on single wavelength bi-directional multiplex. In *Fiber Optics in Access Network (FOAN), 2015 International Workshop on* (pp. 11-16). IEEE.
- 14 Troulos, C. (2013). The Impact of Cost and Demand Uncertainty to the Fiber-to-the-Home Business Case. *Fiber and Integrated Optics*, 32(4), 251-267.
- 15 Boone, T., & Franklin, P. H. (2017, January). Cost modeling for customer premises equipment. In *Reliability and Maintainability Symposium (RAMS), 2017 Annual* (pp. 1-5). IEEE.
- 16 Lang, E., Redana, S., & Raaf, B. (2009, June). Business impact of relay deployment for coverage extension in 3GPP LTE-Advanced. In *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on* (pp. 1-5). IEEE.
- 17 Werner, M., Moberg, P., Skillermark, P., Naden, M., Warzanskyj, W., Jesus, P., & Silva, C. (2008, May). Cost assessment and optimization methods for multi-node radio access networks. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE* (pp. 2601-2605). IEEE.
- 18 Werner, M., Moberg, P., & Skillermark, P. (2008, May). Cost assessment of radio access network deployment with relay nodes. In *ICT-MobileSummit 2008 Conference Proceedings*.
- 19 Machuca, C. M., Krauss, S., & Kind, M. (2013, May). Migration from GPON to hybrid PON: Complete cost evaluation. In *Photonic Networks, 14. 2013 ITG Symposium. Proceedings*(pp. 1-6). VDE.
- 20 Koomey, J., Brill, K., Turner, P., Stanley, J., & Taylor, B. (2007). A simple model for determining true total cost of ownership for data centers. *Uptime Institute White Paper, Version, 2, 2007*.
- 21 Leiva, A., Machuca, C. M., Beghelli, A., & Olivares, R. (2013). Migration cost analysis for upgrading WDM networks. *IEEE Communications Magazine*, 51(11), 87-93.
- 22 Chiha Ep Harbi, A., Van der Wee, M., Verbrugge, S., & Colle, D. (2018). Techno-economic viability of integrating satellite communication in 4G networks to bridge the broadband digital divide. In *ITS2018, the 29th European Conference of the International Telecommunications Society* (pp. 1-11).
- 23 Schneir, J. R., & Xiong, Y. (2016). Cost assessment of FTTdp networks with G. fast. *IEEE Communications Magazine*, 54(8), 144-152.
- 24 Yahyaei, M., & Banisi, S. (2010). Spreadsheet-based modeling of liner wear impact on charge motion in tumbling mills. *Minerals Engineering*, 23(15), 1213-1219.
- 25 Elfeki, A. M., & Bahrawi, J. (2015). A fully distributed spreadsheet modeling as a tool for analyzing groundwater level rise problem in Jeddah city. *Arabian Journal of Geosciences*, 8(4), 2313-2325.
- 26 Grossman, T. A., & Ozluk, O. (2010). Spreadsheets grow up: Three spreadsheet engineering methodologies for large financial planning models. *arXiv preprint arXiv:1008.4174*.
- 27 Lawson, B. R., Baker, K. R., Powell, S. G., & Foster-Johnson, L. (2009). A comparison of spreadsheet users with different levels of experience. *Omega*, 37(3), 579-590.
- 28 Caulkins, J. P., Morrison, E. L., & Weidemann, T. (2007). Spreadsheet errors and decision making: Evidence from field interviews. *Journal of Organizational and End User Computing*, 19(3), 1.
- 29 Powell, S. G., Baker, K. R., & Lawson, B. (2008). A critical review of the literature on spreadsheet errors. *Decision Support Systems*, 46(1), 128-138.
- 30 Van der Wee, M., Casier, K., Bauters, K., Verbrugge, S., Colle, D., & Pickavet, M. (2012, April). A modular and hierarchically structured techno-economic model for FTTH deployments Comparison of technology and equipment placement as function of population density and number of flexibility points. In *Optical Network*

Design and Modeling (ONDM), 2012 16th International Conference on (pp. 1-6). IEEE.

- 31 Vannieuwenborg, F., Verbruggpe, S., & Colle, D. (2017). Designing and evaluating a smart cow monitoring. In CTTE-CMI2017, Internet of Things–Business Models, Users, and Networks (pp. 1-8).
- 32 Naudts, B., Verbrugge, S., & Colle, D. (2015, November). Towards faster techno-economic evaluation of network scenarios via a modular network equipment database. In Telecommunication, Media and Internet Techno-Economics (CTTE), 2015 Conference of (pp. 1-8). IEEE.
- 33 Knoll, T. M. Sensitivity Analysis Add-In for Microsoft Excel. (n.d.). Retrieved February 26, 2018, from [http://www.life-cycle-costing.de/sensitivity\\_analysis/](http://www.life-cycle-costing.de/sensitivity_analysis/)
- 34 Oracle Crystal Ball - Technical Information. (n.d.). Retrieved February 26, 2018, from <http://www.oracle.com/technetwork/middleware/crystalball/overview/index.html>